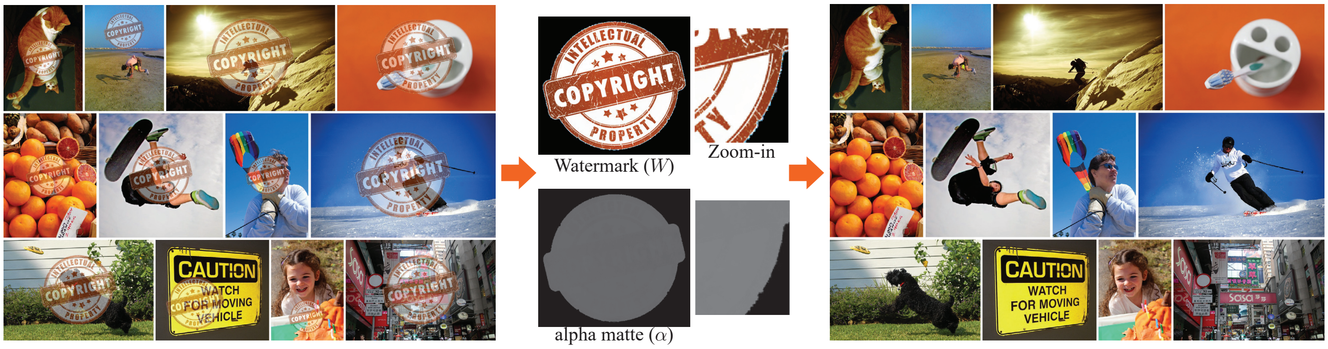
On the Effectiveness of Visible Watermarks

Tali Dekel Michael Rubinstein Ce Liu William T. Freeman
Google Research
{tdekel,mrub,celiu,wfreeman}@google.com



(a) Input watermarked image collection **(b) Computed watermark+ alpha matte** **(c) Recovered images (our result)**

Figure 1. We show that visible watermarks as employed by photographers and stock content marketplaces can be removed automatically. While removing a watermark from a single image automatically is extremely challenging, watermarks are typically added in a *consistent* manner to many images (a). We show that this consistency can be exploited to automatically infer the watermark pattern (b) and to obtain the original, watermark-free content with high accuracy (c). We then investigate and report how robust such an attack is to different types of inconsistencies that may be introduced in the watermarking process to improve its security, such as randomly changing the watermark's position and blend factor, or applying subtle geometric deformation to the watermark when embedding it in each image.

Abstract

Visible watermarking is a widely-used technique for marking and protecting copyrights of many millions of images on the web, yet it suffers from an inherent security flaw—watermarks are typically added in a consistent manner to many images. We show that this consistency allows to automatically estimate the watermark and recover the original images with high accuracy. Specifically, we present a generalized multi-image matting algorithm that takes a watermarked image collection as input and automatically estimates the "foreground" (watermark), its alpha matte, and the "background" (original) images. Since such an attack relies on the consistency of watermarks across image collections, we explore and evaluate how it is affected by various types of inconsistencies in the watermark embedding that could potentially be used to make watermarking more secure. We demonstrate the algorithm on stock imagery available on the web, and provide extensive quantitative analysis on synthetic watermarked data. A key takeaway message of this paper is that visible watermarks should be designed to not only be robust against removal from a single image, but to be more resistant to mass-scale removal from image collections as well.

1. Introduction

Visible watermarks are used extensively by photographers and stock content services to mark and protect digital photos and videos shared on the web. Such watermarks typically involve overlaying a semi-transparent image containing a name or a logo on the source image (Figure 1(a)).

Visible watermarks often contain complex structures such as thin lines and shadows in order to make them harder to remove. Indeed, removing a watermark from a single image without user supervision or a-priori information is an extremely difficult task. However, the fact that watermarks are added in a *consistent* manner to many images has thus far been overlooked. For example, stock content marketplaces typically add similar versions of their logos to previews of many millions of images on the web. We show that the availability of such watermarked *image collections* makes it possible to invert the watermarking process and nearly perfectly recover the images that were intended to be protected (Fig. 1(c)). This can be achieved automatically by only observing the watermarked content.

We show how the problem of watermark removal from an image collection can be formulated as a generalized multi-image matting problem, where the goal is to estimate the "foreground" (watermark) image and alpha matte, along with the "background" (original) images, using many observed examples. Different from natural image matting methods that rely on user scribbles to constraint the problem, our method leverages the redundancy in the data. In particular, we first extract consistent image structures across the collection to obtain an initial estimate of the matted watermark and detect the watermark region in all the images. We then solve an optimization problem that separates the matted watermark into its image and alpha matte components (Fig. 1(b)) while reconstructing a subset of the background images. In our experiments we found that a few

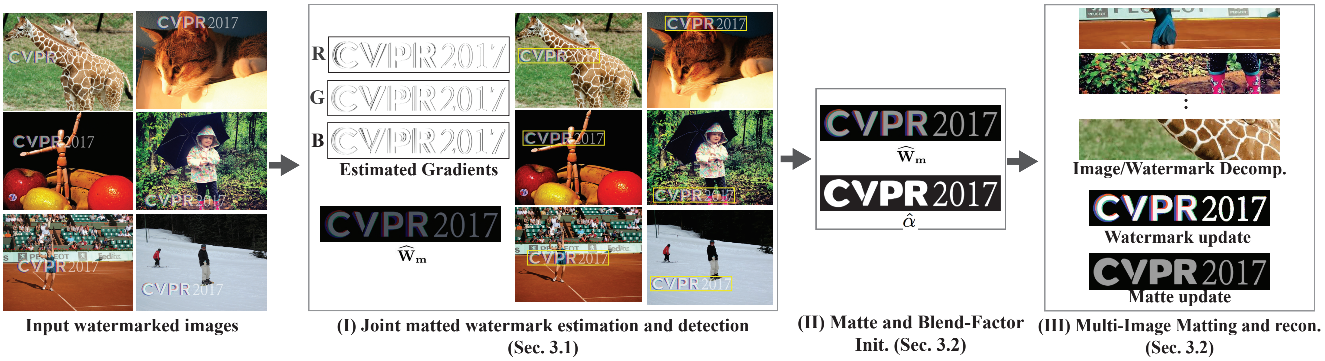

Input watermarked images
R
G
B
Estimated Gradients
$\widehat{W}_m$
(I) Joint matted watermark estimation and detection (Sec. 3.1)
$\widehat{W}_m$
$\hat{\alpha}$
(II) Matte and Blend-Factor Init. (Sec. 3.2)
Image/Watermark Decomp.
Watermark update
Matte update
(III) Multi-Image Matting and recon. (Sec. 3.2)

Figure 2. **Automatic watermark extraction pipeline.** (I) The algorithm first jointly estimates the matted watermark (the product of the alpha matte and watermark image) and localizes it in all the images by detecting consistent image gradients across the collection. This initial estimate is correct up to a spatially-varying shift. (II) The aligned detections are used to estimate an initial alpha matte, and the estimated matted watermark is refined. (III) These are then used as initializations for our multi-image matting optimization.

hundred images marked by the same watermark already suffice for high quality estimation of the watermark and alpha matte. Once the watermark pattern is recovered, it can be efficiently removed in mass scale from *any* image marked by it. Importantly, we do not synthesize or inpaint the watermarked regions; rather, we actually invert the watermarking process to recover the original, watermark-free images.

As such an attack relies on the watermark's consistency across the image collection, a natural question is whether one could prevent it by breaking this consistency. Therefore, we study how robust the attack is to various types of *inconsistencies*—or *variations*—that could potentially be introduced while embedding the watermark in each image. We show, for example, that randomly changing the position of the watermark across the collection does not prevent such an attack from detecting and removing the watermark, nor do random changes in the watermark's opacity or color. Interestingly, we found that applying small spatial deformation to the watermarks during embedding can significantly degrade the quality of the watermark-removed images, with only imperceptible changes to the watermark itself.

We demonstrate results on watermarked image collections obtained from top stock photography web sites, as well as extensive quantitative analysis on synthetic watermarked datasets. A key contribution of our paper is in surfacing vulnerabilities in current visible watermarking schemes, which put many millions of copyrighted images at risk. Specifically, we argue that visible watermarks should be designed to not only be robust against removal from single images, but to be resistant against removal from *image collections* as well. We believe our work can inspire development of advanced watermarking techniques for the digital photography and stock image industries.

2. Related Work

A vast literature exists on digital watermarking (see *e.g.*, [16, 17] for surveys). We focus on *visible* watermarks superimposed on images and limit the scope of our review to work in that area.

Braudaway et al. [3] were among the first to introduce visible watermarks in digital images. They used an adaptive, nonlinear pixel-domain technique to add a watermark to an input image as a means to identify its ownership, while at the same time not obscuring the image details behind it and making the watermark difficult to remove. This scheme has been extended in various ways. Meng and Cheng [13] extended this model to the DCT domain and applied it to compressed video steams. Kankanhalli and Ramakrishnan [8] used statistics of block DCT coefficients to determine the watermark embedding coefficients for each block. They later extended it to account for the texture sensitivity in the human visual system to better preserve the perceptual quality of the images [14]. Hu and Kwong [6] implemented adaptive visible watermarking in the wavelet domain to handle visual discontinuities that may be introduced by DCT-based methods. While some of these methods may improve the visual quality of the watermarks and make them harder to remove, in practice, most modern stock content marketplaces use a standard additive watermarking model, which is the model we focus on in Sec. 3 and generalize in Sec. 4.

As visible watermarking plays an important role in protecting image copyrights, researchers have looked into ways to attack it. Pei and Zeng [15] proposed to use Independent Component Analysis (ICA) to separate the source image from the watermark. Huang and Wu [7] used classic image inpainting methods [2] to fill in the image regions covered by the watermark. These techniques operate on a single image, require a user to manually mark the watermark area and cannot handle large watermarked regions (Fig. 4(b)).

More related to our case are methods for watermark removal in videos [21, 5, 19]. However, such methods rely on temporal coherency of videos, i.e., assume that image content occluded by the watermark in one frame appears unoccluded in others frames [21, 19]. This assumption does not apply to the stock photo collections we deal with. In addition, all these methods inpaint the logo/watermark area, whereas our goal is to explicitly recover the original image by utilizing the semi-visible image content under the watermark.

Watermark removal is also related to classical image matting, where the goal is to decompose a single image into background and foreground layers [9, 18]. As the matting problem is inherently ill-posed and the majority of pixels are either definitely background or foreground, most existing methods rely on a user to provide hard constraints. Moreover, in our setting the opacity of the watermark is typically low in all pixels, i.e., all pixels are either background or mixed. This makes the decomposition more challenging. Finally, natural image matting is typically used for image editing applications such as object cut and paste, which require an accurate alpha matte but can tolerate errors in the background layer. In our case, the quality of the reconstructed background is key. We compare our results with image matting in Sec. 5.

3. An Attack on Watermarked Collections

A watermarked image, J, is typically obtained by superimposing a watermark, W, to a natural image, I. That is,

$$J(p) = \alpha(p)W(p) + (1 - \alpha(p))\, I(p), \tag{1}$$

where $p = (x, y)$ is the pixel location, and $\alpha(p)$ is a spatially varying opacity, or *alpha matte*. The most commonly used watermarks are translucent to keep the underlying image content partially visible [3]. That is, $\alpha(p) < 1$ for all pixels, or $\alpha = c \cdot \alpha_n$, where $c < 1$ is a constant blending factor, and $\alpha_n \in [0, 1]$ is a normalized alpha matte. Similar to natural image matting, for α_n, the majority of pixels are either only background ($\alpha_n(p) = 0$) or only foreground ($\alpha_n(p) = 1$).

Following Eq. 1, and given W and α, one could trivially invert the watermarking process via the per-pixel operation

$$I(p) = \frac{J(p) - \alpha(p)W(p)}{1 - \alpha(p)}. \tag{2}$$

However, when no prior information is available, the problem of recovering I given J alone is extremely challenging and inherently under-determined—there are three unknowns per pixel (W, α, I), and a single constraint (Eq. 1).

However, as discussed in Sec. 1, watermarks are typically added in a *consistent* way to *many* images. Formally, for a collection of images, $\{I_k\}$, marked by the same W and α, we have (omitting the pixel index p for brevity)

$$J_k = \alpha W + (1 - \alpha)I_k, \;\; k = 1, \cdots, K \tag{3}$$

Our goal is to recover W, α and $\{I_k\}_{k=1}^K$ given $\{J_k\}_{k=1}^K$.

This *multi-image matting* problem is still under-determined as there are $3K$ equations and $3(K + 1) + 1$ unknowns per pixel, for K color images. However, the coherency of W and α over the image collection, together with natural image priors, allow solving it to high accuracy, fully automatically.

Our watermark removal algorithm consists of several steps, illustrated in Fig. 2. We next describe each of them in detail. We first consider the case of a consistent watermarking scheme, i.e., the images are marked by the same watermark and alpha matte, in the same position. We then generalize this model in Sec. 4, allowing for positional variations, as well as subtle geometric and color variations across the collection.

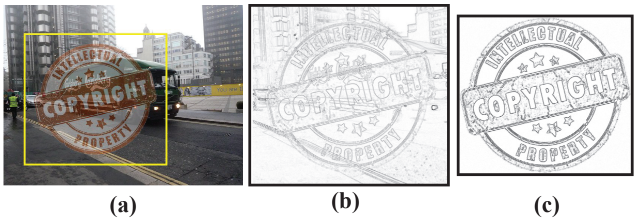


Figure 3. **Initial watermark estimation and detection.** (a) The user provides a rough bounding box around the watermark in a single image (for current stock collections on the web this is not needed; see text). (b) The magnitude of gradients of (a). (c) The magnitude of median gradients across the collection after 2 iterations of watermark detection and estimation (see Sec. 3.1).

3.1. Initial Watermark Estimation & Detection

The first task is to determine which image structures in the collection belong to the common watermark, and to detect them in all the images. This is a chicken and an egg problem since estimating the watermark requires knowing which regions in the images are watermarked, and vice versa. We solve this by jointly estimating the matted watermark and detecting it in all the images. Specifically, we iterate between the following estimation and detection steps.

I. Estimating the Matted Watermark Given a current estimate of the watermarked regions in the images, we determine which image structures in the collection belong to the common watermark by observing consistent image gradients across the collection. Specifically, we compute the median of the watermarked image gradients, independently in x and y directions, at every pixel location p:

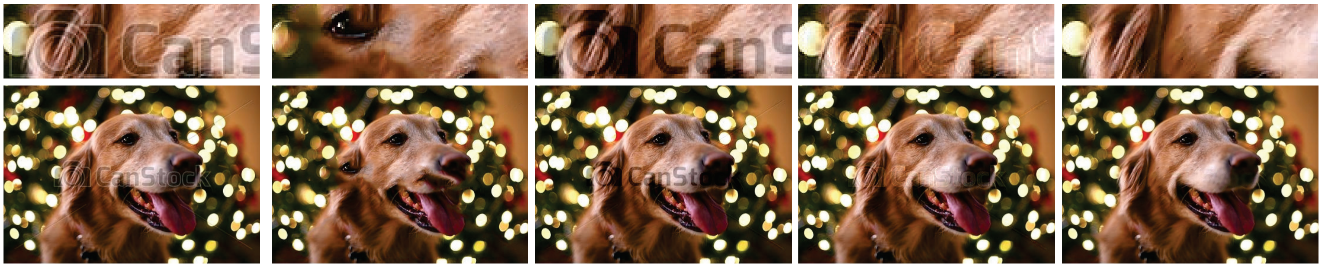
$$\nabla \widehat{W}_m(p) = \text{median}_k(\nabla J_k(p)). \tag{4}$$

As the number of images K increases, Eq. 4 converges to the gradients of the true matted watermark, $W_m = \alpha W$, up to a shift (see Fig. 3). To demonstrate why that is the case, we treat I_k and J_k as random variables, and compute the exception $E\left[\nabla J_k\right]$. Using Eq. 3 we have,

$$\begin{aligned} E\left[\nabla J_k\right] &= E\left[\nabla W_m\right] + E\left[\nabla I_k\right] - E\left[\nabla(\alpha I_k)\right] \\ &= \nabla W_m + E\left[\nabla I_k\right] - \nabla\alpha E\left[I_k\right] - \alpha E\left[\nabla I_k\right] \\ &= \nabla W_m - \nabla\alpha E\left[I_k\right], \end{aligned} \tag{5}$$

where the second equality is from the derivative of multiplication. The third equality is based on the known property of natural image gradients to be sparse, i.e., the chance of having strong gradients at the same pixel location in multiple images is small. Hence, $E\left[\nabla I_k\right] \approx 0$. It follows that $E\left[\nabla J_k\right]$ approximates the gradients of the matted watermark, W_m, except for pixels for which $\nabla\alpha \neq 0$. At those pixels the gradients are shifted by $\nabla\alpha \cdot E\left[I_k\right]$. For now, we continue with this shifted initialization. We will show how this shift can be corrected later on (Sec. 3.2).

We crop $\nabla \widehat{W}_m$ to remove boundary regions by computing the magnitude of $\nabla \widehat{W}_m(p)$ and taking the bounding box



(a) Input **(b) Inpainting** **(c) Matting decomp.** **(d) Direct subtraction** **(e) Ours**

Figure 4. **Watermark removal comparison with baselines.** The watermarked region in (a) is inpainted by photoshop (b) using the estimated matte, α, as a mask. (c) Result by alpha matting decomposition [9]. (d) Result when directly subtracting the estimated watermark (Eq. 2). (e) The result of the attack described in Sec. 3.

of its edge map (using Canny with 0.4 threshold). The initial matted watermark $\widehat{W}_m \approx W_m$ (correct up to a shift) is obtained using Poisson reconstruction (Fig. 3(c)).

II. Watermark Detection Given the current estimate $\nabla\widehat{W}_m$, we detect the watermark in each of the images using Chamfer Distance commonly used for template matching in object detection and recognition [1]. Specifically, for a given watermarked image, we obtain a verbose edge map (using Canny edge detector [4]), and compute its Euclidean distance transform, which is then convolved with $\nabla\widehat{W}_m$ (flipped horizontally and vertically) to get the Chamfer distance from each pixel to the closest edge. Lastly, the watermark position is taken to be the pixel with minimum distance in the map. We found this detection method to be very robust, providing high detection rates for diverse watermarks and different opacity levels, as demonstrated in Sec. 5 and the supplementary material.

To initialize the joint estimation, if the relative position of the watermark in the images is fixed (as is the case with any stock image collection we observed on the web), we get an initial estimation of the watermark gradients, $\nabla\widehat{W}_m$, by registering the images relative to their centers and running step I. If the watermark position is not fixed, we only require the user to mark a rough bounding box around the watermark in one of the images (Fig. 3(a)). We then use the gradients in the given bounding box as an initial estimation for the watermark gradients. We found that iterating between steps I. and II. 2-3 times is enough to obtain accurate detections and a reliable matted watermark estimation.

3.2. Multi-Image Matting and Reconstruction

Given the aligned detections in all input images, our goal is then to solve the multi image matting problem (Eq. 3), i.e., decompose the observed intensities in each image into their watermark, alpha-matte, and original image components. The challenge is how to resolve the inherent ambiguity in this problem completely automatically and reliably.

Our initial estimation of the matted watermark $\widehat{W}_m$ provides us with valuable information about the structures that do not belong to the original images. However, from Eq. 3 it is clear that it is insufficient to constrain the problem as the matted watermark needs to be decomposed into its image component W, and alpha matte component α. Accurate estimation of each component is vital, as very small errors (as little as 2 intensity levels in the watermark) may already show up as visible artifacts in the reconstruction (see Fig. 4(d)).

To address these challenges, we formulate the watermark inversion problem as an optimization problem that jointly solves for W, α and a collection of K watermark-free images $\{I_k\}$. Formally, we define the following objective:

$$\arg\min_{W,\alpha,\{I_k\}} \sum_k \Big(E_{\text{data}}(W,\alpha,I_k) + \lambda_I E_{\text{reg}}(\nabla I_k) \Big) + \lambda_w E_{\text{reg}}(\nabla W) + \lambda_\alpha E_{\text{reg}}(\nabla\alpha) + \beta E_f(\nabla(\alpha W)). \quad (6)$$

The term $E_{\text{data}}(W,\alpha,I_k)$ penalizes for deviations of the k^{th} image from the formation model (Eq. 1), at every pixel p, and is given by

$$E_{\text{data}}(I_k,W,\alpha) = \sum_p \Psi\left(|\alpha W + (1-\alpha)I_k - J_k|^2\right), \quad (7)$$

where $\Psi(s^2) = \sqrt{s^2+\epsilon^2}$, $\epsilon = 0.001$ is a robust function that approximates L_1 distance (p is omitted for brevity).

The terms $E_{\text{reg}}(\nabla I)$ and $E_{\text{reg}}(\nabla W)$ are regularization terms that encourage the reconstructed images and the watermark to be piecewise smooth where the gradients of the alpha matte are strong. We define $E_{\text{reg}}(\nabla I)$ as

$$E_{\text{reg}}(\nabla I) = \sum_p \Psi(|\alpha_x|I_x^2 + |\alpha_y|I_y^2), \quad (8)$$

where I_x, I_y are the horizontal and vertical derivatives of the image, respectively, and Ψ as defined above. The term $E_{\text{reg}}(\nabla W)$ is defined similarly. The regularization term on alpha is given by $E_{\text{reg}}(\nabla\alpha) = \sum_p \Psi(\alpha_x^2 + \alpha_y^2)$.

Even with the use of multiple images and smoothness priors the decomposition problem is still ambiguous. For example, for a fixed alpha matte, there may be infinite number of piecewise smooth watermark/images decompositions that can satisfy the formation model (Eq. 3) . The last term in the objective–*the fidelity term*–reduces this ambiguity by encouraging the matted watermark $W_m = \alpha W$ to have similar gradients to the initial estimate $\widehat{W}_m$, and is given by

$$E_f(\nabla W_m) = \sum_p \Psi(\|\nabla W_m - \nabla\widehat{W}_m\|^2). \quad (9)$$

The impact of the fidelity term is demonstrated in Fig. 5(a), where we compare our result with (top) and without (bottom) this term. In addition, we compare it to the matting decomposition approach taken in [9], using the ground truth alpha matte as input (Fig. 4(d)).

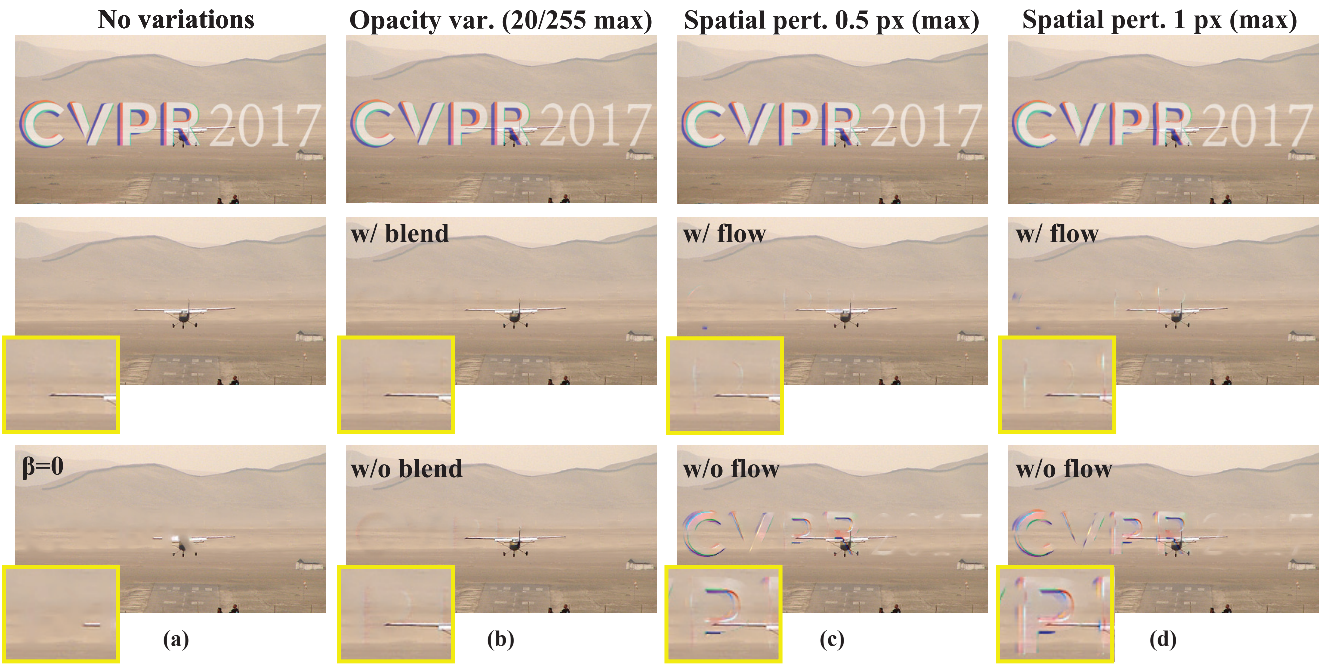


Figure 5. **Robustness to watermark variations.** This figure is best viewed on a monitor. Top row shows the input. (a) Our result for the traditional (consistent) watermarking model; bottom image shows the result for $\beta = 0$, i.e., when the fidelity term (Eq. 9) is not used. (b) Our result when random opacity changes are introduced in each image, with (middle) and without (bottom) estimating a blend factor per image (Sec. 4). Notice no significant visible difference comparing (a) middle and (b) middle. (c) Our results for 0.5 pixel per-image perturbation, with (middle) and without (bottom) flow estimation. (d) similar to (c), for 1 pixel perturbation. Notice how visual artifacts gradually increase with perturbation magnitude, making it difficult for the attack to produce artifact-free reconstruction.

Optimization The resulting optimization problem (Eq. 6) is non-linear and the number of unknowns may be very large when dealing with a large collection ($O(KN)$ unknowns, where N and K are the number of pixels per image, and number of images, respectively). To deal with these challenges, we introduce auxiliary variables $\{W_k\}$, where W_k is the watermark of the k^{th} image. Each per-image watermark W_k is required to be close to W. Formally, we rewrite the objective as follows

$$\arg\min \textstyle\sum_k \big(E_{\text{data}}(I_k, W_k, \alpha) + \lambda_I E_{\text{reg}}(\nabla I_k) + \lambda_w E_{\text{reg}}(\nabla W_k) + \lambda_\alpha E_{\text{reg}}(\nabla \alpha) + \beta E_f(\nabla(\alpha W_k)\big) + \gamma \textstyle\sum_k E_{\text{aux}}(W, W_k), \quad (10)$$

where $E_{\text{aux}}(W, W_k) = \sum_p |W - W_k|$.
Using these auxiliary variables, we solve smaller and simpler optimization problems (using alternating minimization). The resulting iterative algorithm consists of the following steps.

I. Image–Watermark Decomposition At this step, we minimize the objective w.r.t. W_k, and I_k, while keeping α and W fixed. Thus, the optimization in Eq. 10 reduces to:

$$\arg\min_{W_k, I_k} E_{\text{data}}(I_k, W_k) + \lambda_I E_{\text{reg}}(\nabla I_k) + \lambda_w E_{\text{reg}}(\nabla W_k) + \beta E_f(\nabla(\alpha W_k)) + \gamma E_{\text{aux}}(W, W_k). \quad (11)$$

We solve this minimization problem using Iteratively-Reweighed-Least-Square (IRLS), where the resulting linear system is derived in Supplementary Materials (SM).

II. Watermark Update In this step, we opt to estimate a global watermark W that is consistent with all the estimated per-image watermarks $\{W_k\}$ by minimizing the term E_{aux}. This step reduces to taking the median of $\{W_k\}$. That is, $W = \text{median}_k W_k$.

III. Matte Update. Here, we solve for α, while keeping the rest of the unknowns fixed. In this case, we minimize the following objective over α:

$$\textstyle\sum_k E_{\text{data}}(\alpha, I_k, W) + \lambda_\alpha E_{\text{reg}}(\nabla \alpha) + \beta E_f(\nabla(\alpha W)). \quad (12)$$

Here too, the solution is obtained using IRLS (the final linear system is derived in the SM).

These steps are iterated several times until convergence.

Matte and Blend Factor Initialization: The matte in our formulation is given by $\alpha = c \cdot \alpha_n$, where c is a constant blending factor, and α_n is a normalized matte. The initialization for α_n is obtained by first running single image matting [9]. To avoid the required user input of [9], our method automatically computes foreground/background masks (using adaptive threshold on $\widehat{W}_m$) and uses them as "scribbles". The initial matte is taken to be the median over all single-image mattes.

We infer the blend factor from "black" patches (average intensity below 0.01) in the image collection because in those patches the formation model reduces to: $J_k = c \cdot \alpha_n W$ (no image component). Since W is unknown at this point, we use the initial matted watermark (Sec. 3.1) to estimate c. From Sec. 3.1, we have: $\widehat{W}_m = c \cdot \alpha_n W - c \cdot \alpha_n E[I_k]$. Combining these two observations, we get: $J_k = \widehat{W}_m + c \cdot \alpha_n \cdot E[I_k]$. We estimate $DC = E[I_k]$ as the median intensity across the image collection at each I_k'th patch location, and plug our initial estimation of α_n. We then solve for c using least squares over all black patches. In all the datasets we have tested, thousands of dark patches were detected, which is more than enough for a robust estimation of c. Finally, the estimated matte is used to correct for the shift in $\widehat{W}_m$. This is done by adding the term $c \cdot \alpha_n \cdot DC$ to $\widehat{W}_m$ to get the final matted watermark estimation we will use in the next section (Fig. 2(II)).

3.3. Removing the Watermark in a New Image

Once we have the solution for W and α, we can use them to remove the watermark from any new image marked by it, without the need to run the entire pipeline again. As discussed in Sec. 3.2, very subtle errors in the watermark or alpha matte are likely to show up as noticeable visual artifacts. Therefore, we avoid direct reconstruction and instead perform the *image-watermark decomposition* step of our multi-image matting algorithm (Eq. 11).

4. A Generalized Watermarking Model

The consistent watermarking model (Eq. 3) is the one used in practice in every stock image collection we encountered on the Web (see Sec. 5.1). However, a natural question is whether one can avoid the attack by breaking the consistency of the watermark across the collection. To gain insights, we explore the impact of variations in the watermark and alpha matte from image to image.

We assume that watermarks cannot be arbitrary altered, since various design principles and artistic choices are taken into account in generating and placing them in images. Thus, we focus on *subtle variations* that roughly preserve the original appearance of the watermark. Specifically, we generalize our model to allow for two types of variations: (i.) subtle opacity changes (ii.) subtle spatial perturbations (deformations). The generalized formation model for the k'th image is given by

$$J_k = c_k \alpha(\omega_k) W(\omega_k) + (1 - c_k \alpha(\omega_k)) I_k, \qquad (13)$$

where $\omega_k = (u_k(p), v_k(p))$ represents a dense warp field (applied to both W and α), and c_k is the per-image blending factor (that controls the opacity).

Plugging Eq. 13 into our objective function (Eq. 6), and adding regularization term on the warping field leads to

$$\arg\min_{w} \sum \Psi(|J - cW_m(\omega) - (1 - c\alpha(\omega))I|) + \lambda_\omega \sum \Psi(|\nabla \omega|),$$

where $W_m = \alpha W$ (the image index k is omitted for brevity). Here too, we use alternating minimization, i.e., our multi-image matting algorithm has two additional steps: blend factor estimation (solving for c_k), and α-aware flow estimation (solving for ω_k), per image.

Blend factor: The initial estimation from Sec. 3.2 yields an average blend factor. We then solve for a small deviation per image from that estimation. See the supplementary for the full derivation.

Datasets	# Images	Long Edge	Watermak Res.(w×h)	Pipeline (min)	Removal (sec)
AdobeStock	422	1000	623 × 134	33	18
123RF	1376	650	650 × 433	115	60
CanStock	3000	450	450 × 300	28	12
fotolia	285	500	199 × 66	5	1.5
CVPR17	1000	640	403 × 67	34	5
Copyrights	1000	640	339 × 307	47	40

Table 1. **Datasets and running time.** Running time is given for: (i) Pipeline: full framework, including initial watermark estimation and detection (using all images in the dataset), and multi-image matting (using a subset of 50 images). (ii) Removal: reconstruction of a single image.

α-aware flow: In this step we solve for ω based on the current estimate of W, α, c and I. This results in an algorithm which is similar to conventional optical flow and can be solved using IRLS. To do so, we need to linearize the data term. With an existing estimate ω, our goal is to estimate the optimal increment $d\omega = (du, dv)$. The Taylor expansion results in

$$\begin{aligned} W_m(\omega + d\omega) &\approx W'_m + W'_{mx} du + W'_{my} dv, \\ \alpha(\omega + d\omega) &\approx \alpha' + \alpha'_x du + \alpha'_y dv, \end{aligned}$$

where $W'_m = W_m(\omega)$, $\alpha' = \alpha(\omega)$, and $\alpha'_x, \alpha'_y, W'_{mx}, W'_{my}$ are the partial derivatives of α' and W'_m, respectively. The linearized data term is given by (omitting the constant blend factor c)

$$\Psi(|W'_m + (1 - \alpha')I - J + (W'_{mx} - \alpha'_x I)du + (W'_{my} - \alpha'_y I)dv|).$$

This can be related to the classical optical flow equation $\|I_t + I_x du + I_y dv\|$, by denoting $I_t = W'_m + (1 - \alpha')I - J$, $I_x = W'_{mx} - \alpha'_x I$, and $I_y = W'_{my} - \alpha'_y$. We modified the optical flow code [11] to implement this α-aware flow.

5. Results

We tested our algorithm extensively on watermarked image collections by well-known stock photography web sites, as well as watermarked datasets we generated. For each of the datasets, the number of images, long edge, and the watermark resolution are reported in Table 1.

We set the parameters of the multi-image matting algorithm empirically: $\lambda_I \in [0, 1]$, $\lambda_\alpha = 0.01$, $\lambda_W \in [0.001, 0.1]$ $\beta \in [0.001, 0.01]$. In all the experiments, we used a subset of 50 images from the entire collection and 4 iterations, and reconstructed all the watermarked images. We used all available images in each dataset to get the initial estimation of the watermark. We show the effect of the number of images on the performance in the supplementary material. Running times are reported in Table 1 using a non-optimized MATLAB implementation.

5.1. Results on Stock Imagery

We crawled publicly available preview images from popular stock content web sites using 15 predefined queries such as "fashion", "food", "sports" and "nature". Sample results are shown in Fig. 6 (more are available in the SM).

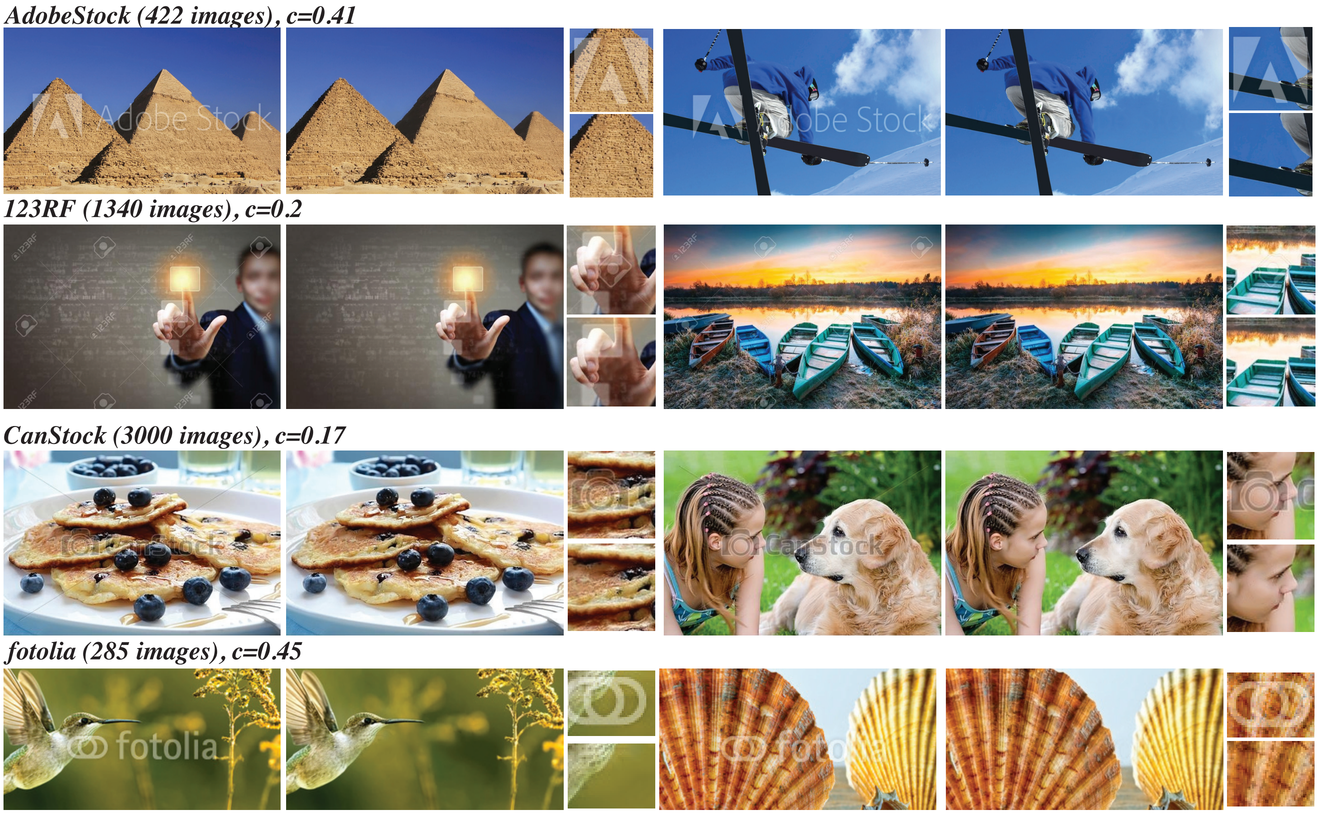

AdobeStock (422 images), c=0.41
123RF (1340 images), c=0.2
CanStock (3000 images), c=0.17
fotolia (285 images), c=0.45

Figure 6. **Results on stock imagery**. This figure is best viewed on a monitor. For each dataset (row), we show (from left to right) the input watermarked image (cropped for better viewing), our automatic, watermark-free image reconstruction, and zoomed in regions. The number of images and estimated blend factor are reported above each row. The corresponding estimated watermarks are shown in Fig. 7)

As can be seen in the input images in Fig. 6, the watermarks contain various structures and shapes, some more complex than the others, including both thick (e.g., *CanStock*) and thin letters and lines (e.g., *123RF*), smooth borders (e.g. *fotolia*), and shadows (e.g. *AdobeStock*). In part of the images, the watermarked region is highly textured, while in others it is smooth. The opacity of the watermarks is low and varies across the different datasets (the estimated blending factors are shown above each row). In all these cases, our algorithm accurately estimated the watermarks (Fig. 7) and original images (Fig. 6).

5.2. Synthetic Datasets & Quantitative Evaluation

We generated a number of watermarked collections, using two watermark images: *CVPR17*, and *Copyrights*. For the source, watermark-free images, we used 1000 images chosen randomly from the Microsoft COCO 'val2014' dataset [10]. With the ground truth in hand, we quantitatively evaluated different aspects of the method using the following metrics:

Detection: We use the Euclidean distance between the center of detected bounding box (Sec. 3.1) and the ground truth; a distance larger than 2px in considered as miss detection.

Reconstruction: We use two well-known metrics: Peak-Signal-to-Noise-Ratio (PSNR), and Structural dissimilarity Image Index (DSSIM), both of which were shown to capture perceptible image degradations. Formally, the DSSIM between each of the reconstructed images $\tilde{I}_k$ and the ground truth I_k is defined as $\text{DSSIM}(\tilde{I}_k, I_k) = \frac{1}{2}(1 - \text{SSIM}(\tilde{I}_k, I_k))$, where $\text{SSIM}(x, y)$ is the Structural Similarity [20]. The total error was measured as the 95% percentile of the DSSIM index map for each image, and taking the mean over the entire image collection.

5.2.1 Consistent Watermark Collection

In *CVPR17/Copyright-fixed*, the images were consistently watermarked using Eq. 3. Sampled input images and results are shown in Figs. [1,2,4,5]. As can be seen (e.g., Fig. 1(b)), our algorithm accurately estimates the fine structures and subtle gradients in the watermark and alpha matte. The computed errors are shown in the Table 2, and Table 3. As a baseline, we evaluated the image reconstructions obtained by standard image matting. It is important to note

	Copyrights-fixed		CVPR2017-fixed	
	PSNR	DSSIM	PSNR	DSSIM
Ours	**36.2**	**0.038**	**32.73**	**0.07**
Matting [9]	15.66	0.46	21.37	0.36
Direct Sub	30.89	0.080	30.65	0.085

Table 2. **Reconstruction quality and comparison**. PSNR and DSSIM (mean over the dataset) for our reconstruction; matting decomposition [9] supplied with the ground truth α; direct subtraction (Eq. 2) with the initial matted watermark.

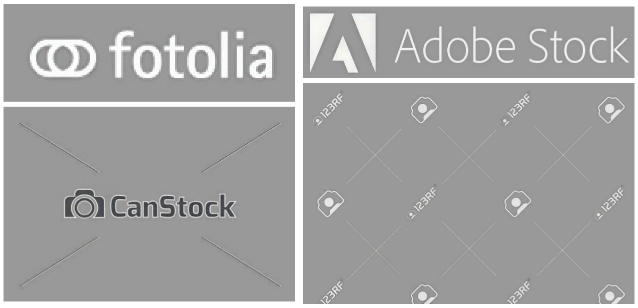


Figure 7. **Estimated watermarks** (αW) for the datasets in Fig. 6.

that most of existing matting methods do not output (nor evaluate) the quality of the underlying background image. To facilitate the task of foreground-background decomposition, we supplied the matting algorithm [9] with the ground truth alpha matte. While this method was able to get reasonable reconstruction in some local regions, it generally fails to resolve the ambiguity between the watermark and background image, especially when the watermark is colored. This can be seen in Fig. 4(c) and the errors in 2.

As a second baseline, we considered the image reconstructions obtained using a direct per pixel subtraction, using our initial matted-watermark. This approach does not generate accurate reconstructions, as small errors in the estimated watermark or alpha matte show up as visual artifacts (Fig. 4(d)).

5.2.2 Robustness to Watermark Variations

We evaluated the robustness of our generalized framework to per-image watermark variations (see Sec. 4). To do so, we generated a number of datasets, using the same logos as before. We first uniformly sampled different position for the watermark per image. We introduced subtle opacity variations by uniformly sampling a blend factor c_k for each image within a few intensity levels around the global blend factor c, i.e., $c_k \in [c - x/255, c + x/255]$ (we used $x = 10, 20$). We generated small spatial perturbations by smoothing two i.i.d random noise images (for the x and y components of the perturbation) with a Gaussian filter. We limit the maximum perturbation in each direction to a defined value (we used a maximum of 0.5 and 1 pixels). We then used those as displacement fields to warp the original watermark and alpha matte (using bilinear interpolation). Finally, we also generated datasets with a combination of the above variations.

We report the results in Table 3. As can be seen, the detection is very robust to various variations. As the *CVPR17* logo is mostly untextured and does not contain strong gradients, it is more challenging to detect, yet its detection rate is still high. We further observe that opacity changes do not affect the results much, and that geometric perturbations have the most significant impact on the quality of the reconstructions. Note that the perturbations do not prevent the algorithm from extracting a reliable estimate of the watermark (as geometric noise can still be integrated out over many images). Therefore, our generalized framework manages to improve the reconstruction quality to some degree (see Fig. 5(c-d)) (top)); however, it is unable to align the perturbed watermark accurately enough and visual artifacts are still noticeable (see Fig. 5(c-d) bottom).

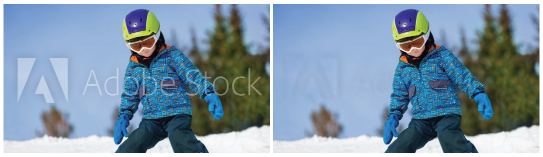


Figure 8. **Example limitation.** Inaccuracies in estimating subtle watermark structures, *e.g*. shadows, may show up as visible artifacts, especially in smooth regions.

6. Conclusion

We revealed a loophole in the way visible watermarks are used, which allows to automatically remove them and recover the original images with high accuracy. The attack exploits the coherency of the watermark across many images, and is not limited by the watermark's complexity or its position in the images. We further studied and evaluated whether adding small random variations in geometry/opacity to the watermark can help prevent such an attack. We found the attack is most affected by geometric variations, which can provide an effective improvement in watermark security compared to current, traditional watermarking schemes.

Fig. 8 shows an example limitation of the attack. In particular, inaccuracies in estimating subtle watermark structures occasionally show up as visible artifacts when the underlying image is smooth. We conjecture that it may be possible to leverage this fact, in addition to the variations, for content-aware watermark placement [12], to further improve robustness to removal.

Dataset	Detection Rate	PSNR	DSSIM
CVPR17			
No Var.	98.6	32.73	0.073
Translation	93.5	32.80	0.087
Opacity (10/255 max)	98.6	32.25	0.062
Opacity (20/255 max)	98.6	31.75	0.066
Spatial pert (0.5px max)	98.0	33.4	0.076
Spatial pert (1px max)	97.8	32.09	0.096
Trans.+Opacity10+pert1	92.4	30.81	0.097
Copyright			
No Var.	100	36.2	0.038
Translation	100	36.35	0.039
Opacity (10/255 max)	100	34.79	0.037
Opacity (20/255 max)	100	33.17	0.063
Spatial pert (0.5px max)	100	33.20	0.059
Spatial pert (1px max)	100	31.23	0.085
Trans.+Opacity10+pert1	100	31.33	0.11

Table 3. **Robustness to watermark variations**. Detection rate (over all images), PNSR and DSSIM (mean over 50 images) for watermarked datasets we generated with several types of random, per-image watermark variations: translation, opacity, geometric perturbation, and their combination. See explanation of variations and magnitudes in the text.

References

[1] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[3] G. W. Braudaway, K. A. Magerlein, and F. C. Mintzer. Protecting publicly available images with a visible image watermark. In *Electronic Imaging: Science & Technology*, pages 126–133. International Society for Optics and Photonics, 1996.

[4] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[5] M. Dashti, R. Safabakhsh, M. Pourfard, and M. Abdollahifard. Video logo removal using iterative subsequent matching. In *AISP*, 2015.

[6] Y. Hu and S. Kwong. Wavelet domain adaptive visible watermarking. *Electronics Letters*, 37(20):1219–1220, 2001.

[7] C.-H. Huang and J.-L. Wu. Attacking visible watermarking schemes. *Multimedia, IEEE Transactions on*, 6(1):16–30, 2004.

[8] M. S. Kankanhalli and K. Ramakrishnan. Adaptive visible watermarking of images. In *Multimedia Computing and Systems, 1999. IEEE International Conference on*, volume 1, pages 568–573. IEEE, 1999.

[9] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):228–242, 2008.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer, 2014.

[11] C. Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, MIT, 2009.

[12] A. Lumini and D. Maio. Adaptive positioning of a visible watermark in a digital image. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 2, pages 967–970. IEEE, 2004.

[13] J. Meng and S.-F. Chang. Embedding visible video watermarks in the compressed domain. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 474–477. IEEE, 1998.

[14] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli. A dct domain visible watermarking technique for images. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 2, pages 1029–1032. IEEE, 2000.

[15] S.-C. Pei and Y.-C. Zeng. A novel image recovery algorithm for visible watermarked images. *Information Forensics and Security, IEEE Transactions on*, 1(4):543–550, 2006.

[16] C. I. Podilchuk and E. J. Delp. Digital watermarking: algorithms and applications. *Signal Processing Magazine, IEEE*, 18(4):33–46, 2001.

[17] V. M. Potdar, S. Han, and E. Chang. A survey of digital image watermarking techniques. In *Industrial Informatics, 2005. INDIN'05. 2005 3rd IEEE International Conference on*, pages 709–716. IEEE, 2005.

[18] J. Wang and M. F. Cohen. *Image and video matting: a survey*. Now Publishers Inc, 2008.

[19] J. Wang, Q. Liu, L. Duan, H. Lu, and C. Xu. Automatic tv logo detection, tracking and removal in broadcast video. In *ICMM*, 2007.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[21] W.-Q. Yan, J. Wang, and M. S. Kankanhalli. Automatic video logo detection and removal. *Multimedia Systems*, 2005.